Wide Area Information Server Concepts

TMC-202

Brewster Kahle
11/3/89
Thinking Machines

Wide Area Information Servers answer questions over a network feeding information into personal workstations or other servers. As personal workstations become sophisticated computers, much of the role of finding, selecting, and presenting can be done locally to tailor to the users interests and preferences. This paper describes how current technology can be used to open a market of information services that will allow user's workstation to act as librarian and information collection agent from a large number of sources. These ideas form the foundation of a joint project between Apple Computer, Thinking Machines, and Dow Jones. This document is intended for those that are interested in the theoretical concepts and implications of a broad-based information system.

The paper is broken up in three parts corresponding to the three components of the system: the user workstation, the servers, and the protocol that connects them. Whereas a workstation can act as a server, and a server can request information from other servers, it is useful to break up the functionality into client and server roles. A final section in the appendix outlines related systems.

Ideas for this have come from Charlie Bedard, Franklin Davis, Tom Erlickson, Carl Feynman, Danny Hillis, the Seeker group, Jim Salem, Gitta Salomon, Dave Smith, Steve Smith, Craig Stanfill, and others. I am acting as scribe. Comments are welcome (brewster@think.com).

I. Introduction

Distributing knowledge was first done with human memory and oral tradition, later by manuscript, and then by paper books. While paper distribution is still efficient distribution mechanism for some information, electronic transmission makes sense for other. This project attempts to install an electronic "backbone" for distribution of information. Some information is already distributed electronically whether it is printed before it is consumed or not. This project attempts to make electronic networks the distribution technique for more types of information by exploiting new technology and standardizing on an information interchange protocol.

The problems that are being addressed in the design of this system include human interface issues, merging of information of many sources, finding applicable sources of information, and setting up a framework for the rapid proliferation of information servers. Accessing private, group, and public information with one user model implemented on personal workstations is attempted to allow users access to many sources without learning specialized commands. A system for finding information in the sea of possible sources without asking every question of every source can be accomplished by searching descriptions of sources and selecting the sources by hand.

An open protocol for connecting user interfaces on workstations and server computers is critical to the expansion of the available information servers. The success of this system lies in a "critical mass" of users and servers. This protocol, then, could be used on any electronic network from digital networks to phone lines.

For the information owners to make their data available over a server, they must be easy to start, inexpensive to operate, and profitable. One possible approach would be to provide software at a low price that will help those with information holdings to put their data on an electronic network. The power of the current personal workstations is enough to enable sophisticate information servicing capabilities. Charging for services can be done in a number of ways that do not entail setting up large billing operations. In this way, it is easy to set up, operate, and charge for information services.

The key ideas that the WAIS system are that information services should be easily and freely distributed, that the power of the current workstations can provide sophisticated tools as servers and consumers, and that electronic networks should be exploited to distribute information.

measure is to count the number of words in common between the question and the text. This well known technique of Information Retrieval[1] can be augmented with different weighting schemes for different words or constructions. Other types of information might be retrieved with specific question formats.

Thus, documents can be found by asking the "navigator" for documents that contain a set of words. Those documents that share the most words with the question will come back at the top of the list (have the best "score"). In this system the "answer" to a question is not a single document, rather it is an ordered list of candidate documents.

Content navigation is not new; NeXT and Lotus have implemented systems for personal computers,[2] many text database systems on mini-computers, and the DowQuest system using a super-computer. In general, there is no standardization yet on how these systems should be queried and used.

B. Dynamic Folders Find Information for the User

Content navigation takes a question and returns an ordered list of possibly relevant documents. The question can be further refined by giving feedback as to how relevant the documents were. The results of a question can be seen as cousin to the file folder in that it contains a list of documents. In reality, the answers to a questions might not be a "copy" of a document, but a "reference" or pointer to a document. These question and answer sessions can be saved just like a file folder can be saved. Saving a session also frees the machine to find answers when the user in not looking. This capability becomes important when some of the questions take time to answer because the data might be far away or difficult to answer. This section discusses one way to think of a saved question: a Dynamic Folder.

"Dynamic Folders" are a cross between a database query and a Macintosh folder that can give us great power in defining questions and probing databases. Text database queries respond with a list of pointers to "hit articles", in the form of titles or headlines, that might interest the user. At that point, the entire article can then be retrieved, if desired. A Dynamic Folder, similarly, has a question that is used to retrieve headlines. Further a Dynamic Folder can be saved and viewed later. Since a folder is a also structure that holds documents so that they can be viewed later, a Dynamic Folder is a folder that has a question associated with it.. In that way a dynamic view acts like a database query in collecting pointers to interesting documents and like a folder in that it can be closed and opened at different times.

A Dynamic Folder's question or "charter" acts as instructions to an active agent as to what what should be put in the folder. This charter gives the folder a mission to keep itself full of appropriate pointers to files or documents. This charter might be as simple as "all files on my personal disk that have a .c suffix", or all mail received in the last day.

In some circumstances, it is important for a Dynamic Folder to contain pointers to a *part of a file* rather than to *an entire file*. Treating parts of files as first class documents is important in systems that group many independent

[1] Salton, Gerald. Introduction to Modern Information Retrieval, McGraw Hill. 1989.

[2] NeXT calls theirs the Digital Librarian, and Lotus calls theirs Megellan (sp?).

Showing the source of documents *geographically* was suggested by Tom Erikson of Apple. In this approach, a world map can be used to show areas of interest. This might be a good way to initiate browsing if geographical relevance is an important factor to the user. The number of articles concerning or originating from an area can be displayed conveniently.

Presenting documents like *books on a shelf* is a familiar metaphor to librarians. Information about the age of the book, how frequently it has been used, its size, if it is a picture book or monograph or pamphlet, when it was published (by the age of the font) are easily gathered with this presentation. Grabbing a book and looking at it, or looking on the shelves close by are natural reactions in this metaphor. I do not know of any attempts to display information in this way.

Generating a recording of a person reading the top articles can be useful for commuters. With simple skip forward and back capabilities, this might be an effective way to deliver a custom newspaper to someone driving a car. This ideally would be done with a CD player, but a cassette could be used.

The Dynamic Folder is just one possible presentation idea. This area will be an interesting area for research and prototypes.

E. Advantages of Remote and Local Filtering

When a user subscribes to a remote server, the user can get a complete copy of the database unfiltered, or can instruct the server to filter the documents remotely. Printed newspapers are delivered whole whether all of it is relevant or not. With electronic distribution, one can imagine a user asking for all sports articles but not the business articles. A query is a form of filter that works at the server. A broad query will retrieve a large number of documents that can be further filtered on the personal workstation. The system and protocols can handle filtering at either or both ends.

Local filtering can done by the content navigation on the local disk after the documents have been retrieved. The quality of this filtering will depend on the quality of the content navigator on the local workstation. The filtering might be able to use knowledge about the user that is impractical to deliver to a server. Local filtering gives the user the most flexibility, but it could entail too much communication or too much disk space. How much filtering will be done on the local workstation has tradeoffs that must be made on a server-by-server basis. If the filtering is done locally, then the workstation might have a *subscription* to a server that periodically retrieves the newest articles.

Remote filtering can reduce the communications bandwidth as well as possibly offer better filtering. A server can have better filtering capabilities because it can be database specific as opposed to the workstation's navigator that must be quite general. Remote filtering, just like an interactive query, in initiated by using a question.

As communications, storage, and local computation costs change relative to each other, different filtering structures might make sense.

G. Local Scoring of Competing Servers

Since a Dynamic Folder can get its data from many servers, it must merge this data and present it in a meaningful way to the user. While servers that rate other servers can help determine which server's answers should be valued (see the ***ratings section), these servers only rate the server as a whole and not the individual documents. Furthermore, the article could be very good, just not appropriate to the question. One way to order the responses presented to the user could be based on a "score" that is assigned to each response by the server. Each server might, for instance, judge the appropriateness of its response to the question on a scale of 1-10. These lists from multiple sources could be merged in that order (weighted by the ratings of the servers) and presented to the user. Unfortunately, since a server would want its data to be used, it has every incentive to rate all articles with at 10. Thus, determining how much to trust the server's scores will improve the selection of documents presented to the user.

One possible solution to this problem is to have local scores for servers to augment what the server says. Therefore, if a server always says "this answer is worth 10" and the user never finds it useful, then the personal workstation can lower the trustworthiness of that server's estimation of itself. Saying 10 all the time is the equivalent to crying wolf; if it does it too often, then users will stop listening. In such a scenario, then, all responses from that server could be degraded by 30% before it is used to merge in with the other database's responses. On the other hand, other databases may underrate themselves and should be boosted.

This local scoring can be used to indicate a user's satisfaction with a database and could be used by others to help in rating it. Further, this local score could be used to determine if the server is worth subscribing to or keeping its articles in the cache.

H. Budgeting the User's Time and Money

Since the users workstation will be spending the users money to contact some servers, a system of accounting and budgeting must be installed so that users get the most value for their money. The trade-offs of time and money can be tricky to try to represent, so a simple system should be attempted first.

The underlying premise is that the computer knows how much it cost to use different services. This can be easy if a service charges for connect time. If a service is reached with a long distance phone call, however this rate could be difficult. (Maybe a server should be set up that knows how much the phone companies charge for different calls.) Further, if a server charges based on the question, there must be a way for the protocol for limiting the amount spent.

Some queries are going to be very important to happen quickly or they are of no use. Working this into the interface can be tricky.

Ideas towards automatic budgeting are still quite primitive. They involve global limits per month, or limits per Dynamic Folder, etc. Should the workstation enforce the limits? Who can override the limits? We need ideas on this one.

it is entered into a directory (which is just another information server) then an English description of the folder should be included.

An information server is probed by putting it in the sources section of the folder's charter. These servers can be varied in size, content, and location. Using content navigation and Dynamic Folders we have an metaphor for accessing many types of information servers.

B. Examples of Information Servers

Information servers, in the broadest sense, answer questions on a particular subject on some network. Electronic networks have been used for years to distribute information in this way. Some of the servers that are available on local area networks have been:

File serving
Printers
Compute servers (such as supercomputers)
FAX
Mail services and archives
Bboard services
Modem pools
Shared databases
Text searching and automatic indexing
CD-ROM servers
Conferencing
Dictionary lookup
User's locations (finger)
Scanners/OCR
35mm Slide output

A local workstation would keep extra information such as:
(1) locally determined "score" reflecting usefulness
(2) subscription information (if any),
(3) user comments, and
(4) time of last contact.
This information would be used to help determine when and if the server should be contacted, and how the responses should be handled.

Navigating in the sea of servers to find new servers can be done using the content navigation technique. In this way a question on classical music would retrieve documents as well as directory entries. This could be done by storing the directory entries on the local disk (in the cache) and accessing it just like local documents based on the appropriateness of the description. Thus retrieving the document would show all the directory information. In that way, a user that is unaware of a certain server would be presented with a description of that server with a listing of its hits for the current question so that s/he could effectively evaluate its potential value of the server. If the server is added to the list of servers for that viewer, then it would be queried in the future.

Maintaining an up-to-date list of services in the cache naturally falls out of content navigation and Dynamic Folders model because a *directory of services viewer* would have the charter to keep itself up-to-date on directory changes, and can be probed using content navigation. The directory of services viewer would list the remote directory server or servers in the sources slot. That way, the directory is kept locally and is fast to access.

Cost and availability information can help guide the workstation to alert its user to new choices of databases. If a new server appears in the directory that is cheaper than the current server, then it could be suggested as an alternative server. This can be complicated to do well, but the benefits of not having the user cull through new directory listings can warrant work in this direction. As Stewart Brand said, "One of the problems with a market based system is that you are always shopping!" Hopefully, the workstation can do some of the mindless part of comparing servers.

Directories are classically owned and serviced by the communications companies. In this role, the communications company is an unbiased party that profits from the use of the system as a whole. Further, communications companies generally take on a teaching role to get users familiar with the system and aid those with problems. This has been true with AT&T with the telephone, the different phone companies with the 900 numbers, and the Network Information Center for the Arpanet. Whether the communications companies take over this role or not, the directory must be supported by some organization or organizations that profit from the use of the system.

D. Servers that Rate other Servers

With a large number of servers, it would be nice to know which ones are sponsored by crooks, and which ones are gems. The directory of information servers necessarily accepts all applications for inclusion, just as the white pages do. Unlike the white pages, however, is a description (or advertisement) of the server is included which can be misleading with the result that users are charged for contacting fraudulent servers. Some protection can be offered by independent grades specific articles as whether they are important. These grades are similar in many ways to the rating services and might be able to be merged.

A Dynamic Folder might have a charter like: "any article from the front page of the New York Times" which is a command to use what the editor suggests the top articles are. Like the rating services, this can be independent of the sources of the articles and combine the information from multiple sources.

A form of editor server would be if users kept track of their favorite articles and put them in a Dynamic Folder and exported it for others. This way, many favorite servers might emerge and articles could be selected based on friend's suggestions.

Automatically figuring out what the user thought of a document is tricky. Clues as to what the user thought of it are:

(1) how many folders point to it,

(2) if the user read it, how much of it, and for how long,

(3) has the user ever taken any information from it to be used in other documents,

(4) has the user ever referenced it.

This type of information could greatly improve users ability to deal with the flood of available information. Furthermore, throwing away all the thoughts a user has about a document is denying others of that mental effort.

F. Markets and Hierarchies: Using Silicon Valley

Currently there are several online information providers and many online information "brokers". Brokers provide the connections between the workstations and the information providers (such as PC-link and Compuserve). Sometimes these brokers have services of their own such as electronic mail and bulletin board services. These brokers try provide a complete information environment by providing access to servers. This structure forces a new information server to be connected to many brokers to have their product used since many users only use a few brokers.. The airline reservation program Eaasy Sabre, for example, is available on 20 of these broker networks. The approach of WAIS is to have an open system of interconnection between users and servers where the brokers can act as a server, but is not an all encompassing information environment. With an open system we have a "market" of information servers rather than a controlled environment or a "hierarchy"[1] . Such a structure could open up the field to many more servers and more sophisticated front-ends.

A market based approach would only standardize on the interchange formats leaving different companies free to store and service queries in any way deemed efficient. The user interfaces, similarly, are free to evolve to fit users needs. Since the protocol is not "terminal oriented" (as most systems are today), it frees the computers on either side to be sophisticated in serving the user.

Rapid evolution of a technology can happen in a market system if the structure is designed well. As long as the protocols are flexible enough to start with, and a procedure for changing the protocol is established, then the components will evolve independently by companies seeking to gain a competitive edge.

[1] Malone, Thomas. Electronic Markets Electronic Hierarchies, CACM June 1987 ***Check this.

IV. The Protocol's Role in WAIS

> "... they have all one language; and this is only the beginning of what they will do; and nothing that they propose to do will now be impossible for them"
>
> Genesis 11:6

To connect a workstation to a server requires a communication network and a language to talk. The communications network can be anything that allows computers to communicate such as modems, Internet, or digital phone networks. A protocol is the language used to relate questions and receive answers between the workstations and servers. This section describes some of the issues involved in this protocol.

A. Open Protocols Promotes Wider Acceptance

It is important to the success of this system to have an open protocol that allows users to connect with servers. Several models for how to create an open standard have been tried, such as: have a company own it and license it (Adobe, for instance), have a university develop it (X Windows, for instance), have a standards organization bless it (Common Lisp, for instance), and simply make the specification available and declare is open (IBM PC, for instance). Each approach has advantages and disadvantages. The key point is that certain attributes be adhered to.

1. The companies that are developing the protocol must be open to using existing standards, and not feeling that new protocols should be protected.
2. A system for enhancements to the standard should be set up. Standards committees are often used for this.
3. The standard should be able to transmit data in a variety of formats. There are many emerging multi-media standards. A good standard will be able to transmit these information standards.
4. The query part of the protocol should be able to accept different formats of queries. Queries might, eventually, have multimedia expressions. These should be free to evolve with periodic standardization.
5. The query must have some method to transmit cost restrictions and time-outs. It should also be able to handle query forwarding while avoiding circularities.

An idea for a query language is to use English that is restricted by the constructs that are understood by the servers. As systems become more complicated, they can handle more English constructs. In this way, future server systems can get more information from a query and produce more appropriate responses, simpler systems might use the words in the query without parsing the structure of the query. This approach would allow the servers to change, while the not changing the human interface and the protocols. The English language approach has been very successful for untrained users of the Dow Jones DowQuest system.

The overall success of this system largely depends on how well these protocols work and how they are made available. There is a standard that could to learn more about their users and start to contact other machines on their user's behalf, the dangers to privacy are significant. There are technical as well as legal issues involved. This section will cover the technical issues in protecting privacy (any good ref for the legal side?).

There is no easy way to protect a personal workstation if an intruder can get at the keyboard. Since the workstation acts on behalf of the user the potential damage that could be done by a crook at the controls would be worse than is currently possible. Since users will be leaving their computer on all the time so that it can contact servers and be used by other servers, we lose the security of the computer being off at night. One way around this might be to able to turn off input from the user while leaving the computer on to contact servers over the network. If a user knows that she is never around at night or on weekends, then this profile might help lead the system to not trust off hour use and require a password. The assumption so far in personal computers is that the machine stays in a secure physical environment and all protection must be directed to network connections. This is not a safe long term solution, and should be thought through carefully.

Other risks are involved when dealing with networks. There are problems with intruders, spies, and forgers. An intruder will try to read, modify, or destroy data that the user did not intend to leave accessible. Spies will watch the traffic from a user to determine the servers contacted and the content of the messages. A forger will copy password information to act like a different user.

Network intruders can be prevented from reading unwanted data by the user only exporting certain Dynamic Folders to become servers for the outside world. A question is whether we want "group" access as well as "world" access as in the Unix file system or some other layered approach. A Dynamic Folder only contains pointers to information. If the information is on the local disk, should that be accessible by a remote machine? Should those files be protected from being read? If the information came from a remote database, should the requester be required to get it from the source even if a copy is on site? What are the copyright issues here?

Spies can watch communications networks and collect passwords and credit card data if this information is sent in clear text (not encrypted) as well as read the data. A public key system makes sense in this application because the directory information can include a key. Public key systems are those that everyone can lock a message (encrypt) for a recipient, but only the recipient can read it. Presumably the public key system would be used in establishing a connection and a special key for the conversation would be established. Current public key systems are too compute intensive to be used for large volumes of data. A conversation key could be used with DES or some other encryption system that is easier to compute (usrEZ software has a product that runs at 30k characters/second on a MacII). Adoption of such a system early in the WAIS development would ensure that this type of protection is assumed in modern information systems.

Forgers can be foiled with a system of authentication. Authentication is important when the charges are high or when the system is used for ordering goods. One solution is to use a public key signature system that is easy to implement using the public key system (ref the Public Key papers). A signature is passed so that only the sender could have created it.

VI. Related Documents

Blip Culture Hypermedia, Harry Chesley, Apple.

Catalyzing a Market of Wide Area Information Servers, Brewster Kahle.

Wide Area Information Server Demonstration, Brewster Kahle and Charlie Bedard.

Electronic Markets and Electronic Hierarchies, Thomas Malone CACM June 1987.

Introduction to Modern Information Retrieval, Gerald Salton, Cornell. McGraw Hill.

Parallel Free-text search on the Connection Machine, Stanfill and Kahle CACM Dec 1986.

C. NetLib is a free Unix utility for distributing files through the email. Anyone that has access to the servers via electronic mail can make inquiries and file requests. This system currently has about 100 (a guess) collections world-wide and is growing. In 1987, about 10,000 requests per month were serviced. The bulk of the offerings are software programs rather than raw data. Since no charges are made for queries or requests this system is used by academics and researchers. ATT and Argonne labs are supporting this work.

The automatic reply system (remote-machine-to-local-machine rather than remote-machine-to-local-human interface) in NetLib is similar to the WAIS system. WAIS, however, is not centered solely around EMail as a transport layer; it uses the phone system as well for interactive use. Also, WAIS would help find databases that are relevant and handle the queries and requests through a more "user friendly" interface. (For more on NetLib see Distribution of Mathematical Software via Electronic Mail in Communications of the ACM May 1987)

D. Switzerland system Still assessing this system.

E. Lotus and NeXT text system

Both Lotus and NeXT have text searching systems that are similar to Thinking Machine's Dow Jones system, but are based on local data (LAN based). Since disks hold close to 1 gigabyte these days, and the entire CM at Dow Jones holds 1 gigabyte, we are close in scope but not performance. On the other hand, a PC will serve its 20 users adequately and the new daily information can be effectively distributed from Dow Jones and other places. Lotus seems to be getting into the information distribution business and is writing software to process that data locally.

These companies see themselves as critically involved in this area. I believe cooperating with them is in our best interest.

F. Information Brokers

Many companies act as brokers to other information providers. Often these services will offer electronic mail and bulletin boards. These private systems rarely communicate with each other. The systems that I know of are listed below. If anyone has any information on these or other companies, please tell me.

AppleLink(Personal Edition)	1-800-227-6364	getting info
Delphi	1-800-544-4005	getting info
Dialcom, Inc.	1-800-435-7342	
GE Information Services	1-800-433-3683	getting info

This company services the fortune 500 companies with network and processing services using Honeywell and IBM mainframes. They lease lines from ATT and provide an environment for their customers including network services and value added filtering and massaging of data.

GEnie	1-800-638-9636	getting info
IBM Information Network	1-800-IBM-2468 ext 100	
INet 2000/TravelNet	1-800-267-8480	bad number
Inet	1-800-322-INET	
NWI	1-800-624-5916	

Quantum Computer Services since 1985, privately held,
"multimillion dollars" official commodore info service. Has been supported by commodore.